

NAMA : MUHAMMAD TARMIDZI BARIQ

NPM : 51422161

KELAS : 2IA11

TUGAS MERANGKUM M1 – M6

Pemrograman Berorientasi Objek (OOP) adalah paradigma pemrograman yang mengutamakan penggunaan objek—yang memiliki atribut dan perilaku untuk memudahkan pengelolaan program yang kompleks. OOP bermula dari pemrograman terstruktur di tahun 1960-an dan berkembang dari penggunaan bahasa seperti C dan Pascal. Dalam OOP, fokus utama adalah pada objek yang dapat menyelesaikan masalah, bukan pada alur prosedural. Pendekatan ini mendukung pengembangan software yang lebih modular dan mudah dikelola, terutama ketika dipadukan dengan Objek-Oriented Analysis And Design Process (OOAD).

PEMOGRAMAN BERBASIS OBJEK

- Menggabungkan fungsi dan data dalam kelas – kelas atau objek – objek
- Memiliki ciri Encapsulation (pengemasan), Inheritance (penurunan sifat) dan Polymorphism (perbedaan bentuk dan perilaku)
- Struktur program ringkas, cukup dengan membuat Objek dan class lalu bekerja berdasarkan object dan class tersebut.
- Kode program sangat re-usable. object dan class dapat digunakan berkali-kali, sehingga dapat menghemat space memori.
- Efektif digunakan untuk menyelesaikan masalah besar, karena OOP terdiri dari class-class yang memisahkan setiap kode program menjadi kelompok - kelompok kecil, sesuai dengan fungsinya.

PEMOGRAMAN TERSTRUKTUR

- Memecah program dalam fungsi dan data
- Memiliki ciri Sequence (berurutan), Selection (pemilihan) dan Repetition (perulangan)
- Struktur program rumit karena berupa urutan proses dan fungsi-fungsi
- Re-use kode program kurang
- Efektif digunakan untuk menyelesaikan masalah kecil dan tidak cocok untuk menyelesaikan masalah yang rumit, karena nantinya akan kesulitan menemukan solusi permasalahan ketika terjadi error

KARAKTERISTIK OBJEK

- Setiap objek memiliki atribut sebagai status yang kemudian akan disebut sebagai **state**.
- Setiap objek memiliki tingkah laku yang kemudian akan disebut sebagai **behaviour**.

Dalam pengembangan perangkat lunak berorientasi objek, objek dalam perangkat lunak akan menyimpan state-nya dalam variable dan menyimpan informasi tingkah laku (behaviour) dalam method atau fungsi-fungsi/prosedur.

MEMBUAT OBJEK

Untuk membuat objek, bisa menggunakan kata kunci **new** yang digunakan untuk membuat objek baru, selanjutnya menentukan 3 langkah untuk membuat sebuah objek. Yaitu: **mendeklarasikan variable, membuat objek baru (Instansiasi)** dan **pemanggilan konstruktor**.

Contoh program:

```
public class Kendaraan{  
    // Konstruktor Dengan Parameter  
    public Kendaraan(String nama){  
        System.out.println("Nama Kendaraannya Adalah "+ nama);  
    }  
  
    public static void main(String[] args){  
        // Perintah untuk membuat objek jenis  
        Kendaraan jenis = new Kendaraan("Pesawat Terbang");  
    }  
}  
  
// Output = Nama Kendaraannya Adalah Pesawat Terbang
```

LATIHAN STUDI KASUS

Buatlah program menggunakan Java dengan kasus seperti berikut:

Seorang pedagang sate setiap harinya menjual dan menerima pesanan dari beberapa pembeli, Harga pertusuknya adalah Rp.950, suatu hari ada pembeli yang memesan 115 tusuk. Jadi berapa harga sate 115 tusuk yang dibayarkan oleh pembeli?

```
package ilab_2ia;  
import java.util.Scanner;  
import java.text.NumberFormat;  
import java.util.Locale;  
  
/**  
 *  
 * @author ASUS  
 */  
public class ILAB_2IA {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
        Scanner input = new Scanner(System.in);  
        NumberFormat currencyFormatter = NumberFormat.getCurrencyInstance(new Locale(language: "id", country: "ID"));  
  
        // Input data barang  
        System.out.print(s: "Nama barang: ");  
        String namaBarang = input.nextLine();  
  
        System.out.print(s: "Harga barang Rp: ");  
        int hargaBarang = input.nextInt();  
  
        System.out.print(s: "Jumlah barang: ");  
        int jumlahBarang = input.nextInt();  
  
        // Hitung total harga  
        int totalHarga = hargaBarang * jumlahBarang;  
  
        // Output hasil  
        System.out.println(s: "-----");  
        System.out.println("Total harga: " + currencyFormatter.format(number: totalHarga));  
  
        input.close();  
    }  
}
```

INHERITANCE

Pewarisan merupakan sebuah bentuk “penggunaan kembali” (reusability); dimana class baru dibuat dari class yang pernah ada yang (biasanya) ditambah fasilitasnya. Setiap class turunan dapat menjadi class pokok (induk) untuk class turunan yang akan datang. Dalam pewarisan, constructor tidak diwariskan pada class turunannya, kecuali jika digunakan perintah super.

- Pewarisan tunggal (single inheritance) merupakan pewarisan dari satu class pokok (induk).
- Pewarisan ganda (multiple inheritance) merupakan pewarisan dari dua atau lebih class pokok.

Class yang mewariskan disebut dengan superclass / parent class / base class, sedangkan class yang mewarisi (class yang baru) disebut dengan subclass / child class / derived class. Untuk menerapkan inheritance, gunakan statement “extends”.

Contoh untuk memanggil constructor milik superclass-nya :

```
super()  
super(parameter)
```

Contoh untuk memanggil atribut dan method milik superclass-nya :

```
super.namaAtribut  
super.namaMethod(parameter)
```

ENCAPSULATION

Merupakan cara untuk melindungi property (atribut) / method tertentu dari sebuah kelas agar tidak sembarangan diakses dan dimodifikasi oleh suatu bagian program.

No	Modifier	Pada class dan interface	Pada method dan variabel
1	Default (tidak ada modifier)	Dapat diakses oleh yang sepaket	Diwarisi oleh subkelas dipaket yang sama, dapat diakses oleh method-method yang sepaket
2	Public	Dapat diakses dimanapun	Diwarisi oleh subkelasnya, dapat diakses dimanapun
3	Protected	Tidak bisa diterapkan	Diwarisi oleh subkelasnya, dapat diakses oleh method-method yang sepaket
4	private	Tidak bisa diterapkan	Tidak dapat diakses dimanapun kecuali oleh method-method yang ada dalam kelas itu sendiri

Aksesabilitas	public	private	protected	default
Dari kelas yang sama	Ya	Ya	Ya	Ya
Dari sembarang kelas dalam paket yang sama	Ya	Tidak	Ya	Ya
Dari sembarang kelas di luar paket	Ya	Tidak	Tidak	Tidak
Dari subkelas dalam paket yang sama	Ya	Tidak	Ya	Ya
Dari subkelas di luar paket	Ya	Tidak	Ya	Tidak

POLYMORPHISM

Overloading adalah diperbolehkannya dalam sebuah class memiliki lebih dari satu nama function/method yang sama tetapi memiliki parameter/argument yang berbeda.

Contoh Overloading

```
1 public class Overloading {
2     public void Tampil(){
3         System.out.println("I love Java");
4     }
5     public void Tampil(int i){
6         System.out.println("Method dengan 1 parameter = "+i);
7     }
8     public void Tampil(int i, int j){
9         System.out.println("Method dengan 2 parameter = "+i+" & "+j);
10    }
11    public void Tampil(String str){
12        System.out.println(str);
13    }
14
15    public static void main(String a[]){
16        Overloading objek = new Overloading();
17        objek.Tampil();
18        objek.Tampil(8);
19        objek.Tampil(6,7);
20        objek.Tampil("Hello world");
21    }
22 }
```

Overriding method adalah kemampuan dari subclass untuk memodifikasi method dari superclass-nya, yaitu dengan cara menumpuk (mendefinisikan kembali) method superclass-nya. Contoh overriding method dapat dilihat pada subclass "Mobil" yang mendefinisikan kembali method keterangan() dan hapus() dari class "Kendaraan".